IMAGE PROCESSING METHOD, SYSTEM AND APPARATUS, AND STORAGE MEDIUM

BACKGROUND OF THE INVENTION

5

Field of the Invention

The present invention relates to an image processing method and apparatus, an image processing system and a storage medium.

10

Description of the Related Art

Conventionally, there is known an image processing system for generating image data in digital form for printing. Such a system, shown in Fig. 1 is generally

15 used. Fig. 1 is a block diagram of a system for preparing a page layout document, a word processor or graphic document or the like by performing DTP (Desk Top Publishing) by using a host computer 101 to hard-copy output by using a laser beam printer, an ink jet

20 printer or the like.

In Fig. 1, the reference numeral 102 denotes an application which operates in a host computer. As representative samples of the application 102, there are such word processor software as Word (R) from

, 25 Microsoft Corporation and such page layout software as PageMaker(R) from Adobe Corporation. Digital documents

prepared by these software are supplied to a printer driver 103 through an operating system (OS) of a computer (not shown).

The digital document is normally represented as a set of command data representing patterns, characters or the like, which configure one page, and these commands are to be transmitted to the printer driver 103. A series of commands configuring a screen are represented by a language system called PDL (Page Description Language). As a representative example of such PDL, there are GDI(R), PS(R)(Post-Script) or the like.

The printer driver 103 transfers the received PDL command to a rasterizer 105 within a raster image processor 104. The rasterizer 105 develops characters, patterns or the like expressed by the PDL command into a two-dimensional bitmap image for actually printer-outputting. Since the bitmap image becomes an image to completely fill a two-dimensional plane by repeating one-dimensional raster (line), the reference numeral 105 is called "rasterizer". The bitmap image thus developed is temporarily stored in an image memory 106.

The foregoing operations are shown in Fig. 2. A document image 111 displayed on the host computer is transmitted to the rasterizer through the printer driver as a PDL command string 112, and the rasterizer

develops the two-dimensional bitmap image on an image memory 106 as denoted by 113. The image data thus developed is transmitted to a color printer 107. The color printer 107 is provided with a known image

5   forming unit 108 of the electrophotographic type or the ink jet recording type, which forms a visible image on a sheet for print-outputting. In this respect, the image data in an image memory 106 is transferred in synchronism with a request for transfer by a

10   synchronizing signal, a clock signal or a specific color component signal (not shown) which is necessary to operate the image forming unit 108.

                    SUMMARY OF THE INVENTION

15

         In such conventional examples as described above, it is obvious that various problems concerning the image forming unit 108 which is utilized for outputting will arise .

20   For example, a color printer normally forms a color image on the basis of the so-called principle of subtractive color mixture using toner or ink of four colors: cyan (C), magenta (M), yellow (Y) and black (K). On the other hand, when the application of the host

25   computer displays an image, a color monitor is normally used, and the color monitor displays the colors using

three primary colors of additive color mixture: red (R),
green (G) and blue (B). The colors of an image or the
like which have been laid out by reading characters,
patterns, photographs or the like, which configure a

5   document, by a scanner are all represented by colors
obtained by mixing R, G and B at a certain ratio. More
specifically, the rasterizer must generate a bitmap
image to be transmitted to the printer, after
converting the color information which has been defined

10 using R, G and B as PDL and transferred from the host
computer, into C, M, Y and K by some means or other.

The method for converting RGB into CMYK is,
however, not uniquely determined, but the optimum
converting method differs depending upon the attribute

15 of a pattern defined by PDL. Referring to, for example,
the example of Fig. 2, the portion denoted by 114 is a
natural image read by a scanner or the like, portions
denoted by 115 are graphic images electronically
generated such as a circle and a rectangle, the portion

20 denoted by 116 is a character (TEXT) image, and each of
them has a respectively different attribute.

In a case where the color of TEXT of the character
image 116 is defined as black, i.e., R=G=B=0, when the
optimum CMYK signal thereto is represented by a 8-bit

25 density signal, the following equations are
established: C=M=Y=0 and K=255. In other words, a

- 4 -

black character is preferably reproduced only by the black toner of the four color toners of the printer. On the other hand, in a case where the pixel value of a specific pixel of the natural image 114 is R=G=B=0,

5 when it is converted to C=M=Y=0 and K=255 as in the case of the character data, it becomes lower in absolute density than the natural image originally expressed, because reproduction is performed only by black toner when a portion having the highest density

10 of the natural image is expressed. In this case, if it is converted into values such as C=M=Y=100 and K=255 to increase the absolute density, desirable results will be obtained.

As another example, there is a case where the

15 image forming unit can reproduce only binary dots. In this case, the rasterizer develops a multi-value bitmap image of Y, M, C and K on an image memory. On receipt of the multi-value bitmap, the image forming unit performs a known binarization process such as the error

20 diffusion method and dithering, and print-outputs after converting a multi-value image signal into a binary image signal. At this time, the optimum binarization method varies depending upon the attribute of the image.

More specifically, for graphics such as characters

25 and patterns, binarization, in which the matrix size of dithering is made smaller to place emphasis on

resolution, is preferable. Also, for such natural images as photographs, it is preferable to make the matrix size larger for placing emphasis on tonal quality.

5   As described above, the optimum development process using the rasterizer differs depending upon the data attribute of the object, and if rasterizing is performed using a single processing method, the image quality will be degraded.

10   The present invention is achieved in the light of the above-described problems, and is aimed to realize, on print-outputting image data, appropriate data processing in accordance with the attributes of each image contained in the image data for improving the

15   quality of the output image.

According to a preferred configuration of the present invention, an image processing apparatus comprises generation means for generating a bitmap image on the basis of object data inputted; holding

20   means for bringing attribute information representing an attribute of the object data inputted into correspondence with each pixel of a bitmap image generated by the generation means for holding; conversion means for converting the bitmap image

25   generated by the generation means into data capable of being processed by an image output unit; and switching

means for switching the contents of processing in the conversion means on the basis of the attribute information held by the holding means.

Other features and advantages of the present
5   invention will be apparent from the following description taken in conjunction with the accompanying drawings, in which like reference characters designate the same or similar parts throughout the figures thereof.
10

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram showing the configuration of a conventional image processing
15 system;

Fig. 2 is a view for explaining procedure of processes executed in the conventional image processing system;

Fig. 3 is a block diagram showing the
20 configuration of an image processing system according to a first embodiment;

Fig. 4 is a view showing an example of a bitmap image for an object to be processed according to the first embodiment;

25   Fig. 5 is a view for explaining attribute map information according to the first embodiment;

Fig. 6 is a view for explaining attribute map information according to the first embodiment;

Fig. 7 is a view showing an example of dither matrix according to the first embodiment;

Fig. 8 is a view showing an example of dither matrix according to the first embodiment;

Fig. 9 is a view showing an example of dither matrix according to the first embodiment;

Fig. 10 is a view showing an example of an edge emphasizing filter according to the first embodiment;

Fig. 11 is a view for explaining a smoothing process according to the first embodiment;

Fig. 12 is a view for explaining a smoothing process according to the first embodiment;

Fig. 13 is a flow chart showing a processing flow of processes to be executed in the first embodiment;

Fig. 14 is a block diagram showing the configuration of an image processing system according to a second embodiment;

Fig. 15 shows an example of a data format of attribute information according to the second embodiment;

Fig. 16 shows an example of attribute map information brought into correspondence with each pixel;

Fig. 17 shows an example of a method for storing an attribute map when image data and attribute map information are stored in the same storage medium;

Fig. 18 shows an example of a method for storing an attribute map when image data and attribute map information are stored in the same storage medium;

Fig. 19 shows another example of a method for storing an attribute map when image data and attribute map information are stored in the same storage medium;

Fig. 20 shows another example of a method for storing an attribute map when image data and attribute map information are stored in the same storage medium;

Fig. 21 shows an example of dither matrix of size 3 × 3;

Fig. 22 shows an example of dither matrix of size 8 × 8;

Fig. 23 is a flow chart for explaining painting of an object and generation of an attribute map according to the second embodiment;

Fig. 24 is a flow chart for explaining procedure of generation of CMYK data due to an image processing unit 147;

Fig. 25 shows an example of a data format for attribute information according to a third embodiment;

Fig. 26 shows an example of a data format for attribute information according to a fourth embodiment;

Fig. 27 shows another example of a data format for attribute information according to the fourth embodiment;

Fig. 28 is a sectional view showing the configuration of a printing device to which a fifth embodiment is applicable;

Fig. 29 is a conceptual diagram showing functions to be implemented by the fifth embodiment;

Fig. 30 is a conceptual diagram showing functions to be implemented by the fifth embodiment; and

Fig. 31 is a flow chart showing the procedure of the fifth embodiment.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Hereinafter, with reference to the drawings, the detailed description will be made of preferred embodiments according to the present invention.
[First Embodiment]

Fig. 3 is a block diagram showing the configuration of an image processing system according to a first embodiment of the present invention.

In this respect, in Fig. 3, the reference numerals 10 to 15 are the same as the reference numerals 101 to 106, respectively in Fig. 1, and the reference numerals

18 and 19 are the same as the reference numerals 107 and 108 in Fig. 1, respectively.

The present embodiment is characterized by an attribute map memory 16 and an image processing unit 17. In this case, the image processing unit 17 is shown such that it independently exists, but it may be configured such that it is included in a raster image processor 13 or a color printer 18.

A rasterizer 14 generates a bitmap image on an image memory 15 on the basis of commands associated with individual components (hereinafter called "objects") configuring an image. At this time, attribute map information is generated on the basis of the attributes of the objects and the generated bitmap image using a method to be described later, and is written in an attribute map memory 16.

More specifically, the rasterizer 14 generates attribute map information on the basis of the attributes of commands representing the objects and bitmap image data which have been generated and written in the image memory 15 in accordance with the commands. At this time, it is also naturally possible to refer to the contents of the image memory 15 already developed as bitmap image data.

The image processing unit 17 subjects the bitmap image data in the image memory 15 to various image

- 11 -

processing to output data to a color printer 18. At
this time, referring to the attribute map information
in the attribute map memory 16, the image processing
method is appropriately switched.

5   First, detailed description will be made with
respect to a method for generating the attribute map
information.

In a case where bitmap image data developed in the
image memory 15 is as shown in Fig. 4, enlarged

10   diagrams for areas (a) and (b) are shown in FIGS. 5 and
6 respectively.

Fig. 5 shows a bitmap image for a circle which is
generated on the basis of a command for painting a
circular object. The reference numeral 21a denotes

15   bitmap image data to be written in the image memory 15,
and is obtained by arranging pixel values in units of
pixels as an integer value of, for example, eight bits
like a two-dimensional array.

The reference numerals 22a to 25a denote attribute

20   map information obtained by analyzing the bitmap image
data to obtain each attribute thereof and attribute
information flags thereof. The attribute map
information is written in the attribute map memory 16.
The attribute information flags (in the present

25   embodiment, four types: vector flag, character flag,
edge flag and edge boundary flag) are generated, each

being 1 bit (binary of 0 or 1), in the same form of pixel array as the bitmap image data of the image memory 15. In Fig. 5, each flag is represented by minute rectangles having 0 as white and 1 as black.

5   The vector flag is set to 1 in a vector image area such as characters and graphics, and to 0 in any ground (state in which there is no object information) portion and a continuous tonal photograph portion (area (c) in Fig. 4 other than the vector image area). In this case,

10   therefore, data 22a, which is the entire interior of a circular object, has 1 as a vector flag. The data 22a can normally be generated on the basis of a command for painting the circular object. In the present embodiment, it is possible to detect a newly filled

15   area in the bitmap image 21a by referring the contents of the bitmap image 21a, and set vector flags for the detected area to 1 thereby data 22a is obtained.

        The character flag becomes 1 in the character image area, and 0 in all other areas. In this case,

20   therefore, since the circular object is not a character, the data 23a is all 0.

        The edge flag becomes 1 at the boundary portion of the object, and 0 in all other areas. In this case, therefore, data 24a, which is the boundary of the

25   circular object, has 1 as the edge flag. This can be generated by detecting a pixel which varies from 0 to 1

in the data 22a and setting the flag to 1 at the pixel position thus detected.

The edge boundary flag is set to 1 if the corresponding pixel is adjacent to a pixel whoes edge flag has been set to 1. In this case, a pixel which is adjacent in the vicinity of 4 of the pixel whose value is 1 in the data 24a, is detected, whereby values for the pixels located both in the inside and in the outside of the pixel having value "1" in the data 24a are set to 1. Thus, the data 25a indicates an edge boundary flag. Depending upon the contents of the image processing to be described later, however, there are also cases where only the pixel at the outside of the edge has preferably 1 as the edge boundary flag. In this case, it is also possible to prevent any edge boundary flag from being generated in the intermediate density portion (area denoted by gray) in the inside of the circle by referring to the original image memory 21a together with the data 24a.

Fig. 6 shows an example for generation of attribute map information with respect to a character object. 21b to 25b have the same meaning as 21a to 25a, and the attribute map information generated from them is also almost the same. Only data 23b is different from the data 23a, and has 1 as the character flag. This is because since 21b is a character object, the

character flag is set to 1 for the entire interior of the character.

In accordance with the foregoing procedure, the attribute map information is generated. The bitmap image data and attribute map information thus generated are transferred to an image processing unit 17 together with the synchronizing signal. At this time, the data at a predetermined pixel position of the image memory 15 and the attribute map information of the predetermined pixel must be brought into correspondence with each other when they are transferred. More specifically, when the pixel value of a specific pixel in the image memory 15 is transmitted to the image processing unit 17, the attribute map information (flag) of the same pixel must also be transmitted at substantially the same time. In this respect, in the continuous tonal image area shown in the area (c) in Fig. 4 and in the ground area where no image appears, all flags become 0 as will be apparent from the foregoing description.

The image processing unit 17 subjects pixel values from the image memory 15 to various image processing and the processing method is controlled by a flag from the attribute map memory 16.

In other words, with respect to the bitmap image data developed on the image memory 15, for example,

switching of the size of the dither matrix, switching
of a space filter for space filter processing such as
edge emphasizing, switching of such smoothing process
as so-called SST or the like are performed based on the
5   attribute map information.

With reference to Figs. 7 to 9, descriptions will
be made of an example of processing in the case of
switching the size of dither matrix based on the
attribute map information.

10   Figs. 7 to 9 show an example of dither matrix
according to the present embodiment.

It is assumed that the bitmap image data used in
the present embodiment is an integer value (0 to 255)
of eight bits for each pixel. A value within each cell
15   of the dither matrix is used as the threshold value for
binarization, whereby ON/OFF of the output pixel is
controlled. Fig. 7 shows a dither matrix of size 3 × 3,
which is capable of expressing 10 levels of gray in a
resolution of 200 dpi assuming an original image of 600
20   dpi. Fig. 8 shows a dither matrix of size 5 × 5, which
is capable of expressing 120 dpi/26 levels of gray.
Fig. 9 shows a dither matrix of size 8 × 8, which is
capable of expressing 75 dpi/65 levels of gray.

Therefore, referring to the attribute map
25   information corresponding to the objective pixel, in
case of a continuous tonal area, that is, when the

- 16 -

vector flag is 0, the object pixel is binarized using the dither matrix of Fig. 9. In the case of a graphic area, that is, when the vector flag is 1 and the character flag is 0, the object pixel is binarized

5 using the dither matrix of Fig. 8. Also, in case of a character image area, that is, when the character flag is 1, the object pixel is binarized using the dither matrix of Fig. 7. Thus, characters can be reproduced at high resolution while photographs or the like are

10 being reproduced with emphasis placed on the gradation, and graphics can be reproduced on intermediate density level, between them. In this manner, high-quality output images can be obtained even if bitmap images of a plurality of types of attribute map information are

15 mixed.

Also, in the case of a pixel whose edge flag is 1, by referring to the edge flag, prior to the above described dithering, such as an edge emphasizing filter as shown in Fig. 10 is convoluted in space, whereby the

20 sharpness of the characters and graphics can be further improved.

Also, smoothing processing, in which half tone pixels are added to the edge portion of a character or graphic to reduce jaggies of the edge portion, can be

25 also performed. This processing is to add half tone pixels to the outside of the edge portions of, for

example, the image in Fig. 11 so as to make the edge portion as shown in Fig. 12 for suppressing backlash of the edge. This processing is performed only for the pixel having edge boundary flag of 1, whereby more

5　effective results can be obtained.

In Fig. 12, only the pixels in the outside of the edge boundary are processed. This is because the interior of the pattern is not white, and therefore if the same processing is performed also on the inside

10　edge boundary, defects would occur. Therefore, the edge boundary flag is controlled in accordance with the contents of the object in such a manner that the edge boundary flag becomes 1 only in the outside of the edge or the edge boundary flag is set to 0 when the ground

15　is not white.

In the foregoing embodiment, as the attribute map information of an image, the description has been made of four types of attributes: whether or not it is a vector image, whether or not it is a character image,

20　whether or not it is an edge, and whether or not it is the boundary portion of an edge. However, the present invention is not limited thereto. Various flags can be applied, for example, whether the object has a chromatic color or an achromatic color, whether an edge

25　portion is on a ground or on a white ground, or whether or not an edge portion is in a photograph.

- 18 -

Also, switching of an image processing method based on the attribute map information is not limited to the foregoing embodiment, but in addition, there are conceivable various methods such as switching of a

5   color conversion method from RGB signal to YMCK signal, and switching of a compression method when the bitmap image data is compressed to transfer it to the printer unit.

Also, since the bitmap image data already

10 developed is adapted to be referred when a flag is generated, for example, it is also possible to process so as not to set any character flag, in a case where a character is over-written in an area in which a gray-scale image has been already developed.

15 Next, with reference to Fig. 13, a description will be made with respect to a processing flow to be executed in the present embodiment.

Fig. 13 is a flow chart showing a processing flow to be executed in a first embodiment according to the

20 present invention.

First, in a step S101, each object of image data configured by a plurality of inputted objects is developed on the bitmap image data by a rasterizer 14. The bitmap image data thus developed is stored in the

25 image memory 15. In a step S102, attribute map information of each bitmap image data is generated on

the basis of the developed bitmap image data and the attribute of the object. The generated attribute map information is stored in an attribute map memory 16. In a step S103, an image processing method for each bitmap image data is determined on the basis of the generated attribute map information, and executed at an image processing unit 17. Thus, an image based on the image-processed bitmap image data is formed at an image forming unit 18.

As described above, according to the first embodiment, bitmap images, in which bitmap images of a plurality of types of attribute map information are mixed, can be subjected to image processing suitable for the attribute map information, and therefore, an output image based on the bitmap image can be outputted with higher quality.

As described above, according to the first embodiment, it is possible to provide an image processing apparatus capable of outputting an image comprising images of a plurality of types of attributes mixed, with high quality, a method therefor and a computer-readable memory.

[Second Embodiment]

Hereinafter, with reference to the accompanying drawings, the detailed description will be made of a

second preferred embodiment according to the present invention.

Fig. 14 is a block diagram showing the configuration of an image processing system according to the second embodiment.

In Fig. 14, the reference numeral 140 denotes a host computer; and 141, application to be used within the host computer 140. As applications 141, there are, for example, word processor software, drawing system of software, graphic software or the like. Reference numeral 142 denotes a printer driver, which serves as an interface with the printer for outputting characters, graphics and bitmap images from the application 141 to the printer.

The reference numeral 143 denotes a raster image processor for developing output obtained through the printer driver into image data. Within the raster image processor 143, there are contained a rasterizer 144, an image memory 145 for storing the image data therein, and an attribute map memory 146. In this respect, the raster image processor may belong to the host computer or the printer.

The reference numeral 147 denotes an image processing unit, which converts the image data stored in the image memory 145 while referring to the information stored in the attribute map memory 146,

into image data (for example, binary image data each of YMCK) which is outputted by an image forming unit 149 to be described later. The reference numeral 148 denotes a color printer. The color printer 148 includes an image forming unit 149 for printing image data, on a medium such as paper. The image forming unit 149 is a unit of the electrophotographic type, a unit of the ink jet type or the like. The final output result can be obtained through this image forming unit 149.

In this respect, in the present embodiment, a description will be made using a color printer 148 as a printer for performing print output, but it will be apparent that the processing of the present embodiment is easily applicable to a monochrome printer.

Also, in the present embodiment, the image memory 145 and the attribute map memory 146 are represented as a separate block respectively for the sake of understanding, but it is, of course, possible to have one recording medium to store both the image data and the attribute map data.

A digital document prepared in the application is represented by a command system called "PDL" as described in the conventional example. The command is roughly composed of three objects. The first is a character object, the second is a graphic object of

vector data or the like such as patterns and free curved lines, and the third is a bitmap object such as image data obtained by reading photographs or printed matter by a scanner or the like.

5   The object is, if it is a character, for example, composed of data such as a character code for identifying which character it is, a font for defining the shape of a character, size information for representing a size of the character, and color

10 information for representing the color of the character, and cannot be interpreted by the image forming unit as it is. For this reason, the printer driver 142 transmits a command string representing objects such as characters, graphics and bitmaps to the rasterizer 143

15 while serving as an interface with the printer, and synchronizing in such a manner that the image forming unit 149 at the final stage can output an appropriate image.

The rasterizer 143 converts the received command

20 string into two-dimensional bitmap data which can be properly interpreted by the image forming unit, and at the same time, outputs an attribute map. The attribute map is two-dimensional information corresponding to each image pixel of attribute information. This

25 attribute map is stored in the attribute map memory 146 in such a manner that the two-dimensional image data

can be brought into correspondence with the attribute map for each pixel.

Fig. 15 shows an example of a data format for attribute information. In the present embodiment, the attribute information is 1-bit information represented by 0th bit, and the 0th bit represents a bitmap flag. If the 0th bit is 1, it signifies a pixel generated from the bitmap object, and if it is 0, it signifies a pixel generated from the vector object, that is, a character or a graphic image.

The rasterizer 143 judges, on converting an object into two-dimensional bitmap data, for each pixel, from which type of an image, i.e., a character, graphics image or natural image, the object has been generated, and stores the attribute information in the attribute map memory 146 in such a manner that the attribute data can be brought into correspondence with the two-dimensional image data.

Fig. 16 shows an example of attribute map information brought into correspondence for each pixel. Fig. 16 represents an image obtained by placing a numeral "1", which is a character object, on the bitmap object. On converting into a two-dimensional bitmap, such attribute map information as shown in Fig. 16 is generated in accordance with the format described in Fig. 15. In other words, for each pixel if it is a

pixel generated by a vector (character or graphic object), "0" is outputted, if it is a bitmap object, "1" is outputted. Said attribute map, as shown in Fig. 16, is stored in the attribute map memory 146.

5   In this respect, the attribute map may be configured in any manner so long as it is stored such that correspondence can be performed for each pixel. A method, in which attribute information is generated in an attribute map plane is shown in Fig. 16. Image data

10   is stored in the image memory 145, and the attribute map information is stored in the attribute map memory 146. This is one example, and another mode may be adopted.

          If, for example, image data and attribute

15   information are stored in the same storage medium, an attribute map plane may be added to each plane of RGB as shown in Fig. 17. Or, if RGB data is configured for each pixel as shown in Fig. 18, it may be embedded such that it is added to the RGB information for each pixel.

20   Also, in order to configure so as not to increase the amount of data, the configuration may be arranged such that attribute information is embedded in the lower bit for each pixel on any of RGB planes or a plurality of planes thereof as shown in Fig. 19, or that when RGB

25   data is configured within 1 pixel as shown in Fig. 20, the attribute information is embedded in the lower bit

of any of information of eight bits each of RED, GREEN and BLUE for each pixel, or a plurality of color information. In the example of Fig. 20, there is shown an example in which the attribute map has been embedded

5 in the lower 3 bits of 8 bits of information of BLUE (although the attribute information is 1 bit in the second embodiment, 2 bits or 3 bits can be used as the attribute information as described in the third and fourth embodiments later).

10 The image processing unit 147 fetches the attribute map information embedded as described above, determines the attributes of each pixel on the basis of the image data stored in the image memory 145 and the attribute map brought into correspondence with the

15 image data for each pixel, and switches the image processing to perform the optimum image processing for each attribute of the image.

With reference to Figs. 23 and 24, the description will be further made on the procedure in the second

20 embodiment described above.

Fig. 23 is a flow chart for explaining rendering of an object and generation of an attribute map according to the second embodiment. First, in a step S11, the rasterizer 144 analyzes object inputted data.

25 In a step S12, an image based on RGB is developed and is rendered in the image memory 145. Next, in a step

S13, it is judged whether the attribute of the object is a "natural image" or a "graphic or character".

If it is determined to be a natural image in the step S13, the procedure will proceed to a step S14, the attribute map information is set to "1", and is written at positions on the attribute map memory 146 corresponding to each pixel of the object (step S16). On the other hand, if it is determined to be a graphic or a character in the step S13, the attribute map information is set to "0", and is written at positions on the attribute map memory 146 corresponding to each pixel of the object (steps S15 and S16).

In a step S17, it is judged whether or not there is a next object to be rendered, and if present, the procedure will return to the step S11, and if not present, the present processing will be terminated.

Fig. 24 is a flow chart for explaining a procedure for generating CMYK data by the image processing unit 147.

First, in a step S21, the image data is read out from the image memory 145 in units of pixels. In the second embodiment, RGB multi-value data is stored in the image memory 145. In a step S22, the attribute information of pixels read out in the step S21 is read out from the attribute map memory 146. In this respect, as regards storing mode of the attribute information,

various variations are shown in FIGS. 18 to 20, and in the step S22, the attribute information of pixels can be read out using a read method appropriate to the respective modes.

In a step S23, the RGB multi-value data is converted into YMCK multi-value data, which is converted (binarized) by the processes in steps S24 to S26 into binary data for each of YMCK colors which can be processed by the image forming unit 149. In this binarization, a dither matrix is used in the present embodiment, and at this time, the dither matrix is properly used in accordance with the attribute information of each pixel.

First, in the step S24, it is judged on the basis of the attribute information read out in the step S22 whether the attribute of the pixel to be processed is a "natural image" or a "graphic or character". If the attribute of the pixel to be processed is a "graphic or character", the procedure will proceed to the step S25 for binarization using, for example, such a 3 × 3 dither matrix as shown in Fig. 21 (priority is placed on resolution). On the other hand, if the attribute of the pixel to be processed is a "natural image", the procedure will proceed to the step S26 for binarization using a 8 × 8 dither matrix (priority is placed on gradation).

As described above, the following processing can be performed through the use of the attribute map information. Referring to the example of Fig. 4, the respective portions have attributes which are different from one another respectively such that the portion (c) is a natural image read by a scanner or the like, the portion (a) is graphic images program generated such as a circle and a rectangle, and the portion (b) is a character (TEXT) image. Now, let us consider such a case that the image forming unit can reproduce only binary dots.

In this case, the rasterizer 144 develops multi-value bitmap images of Y, M, C and K on the image memory, on receipt of which the image forming unit 149 performs the known binarization process such as the error diffusion method and dithering to convert a multi-value image signal into a binary image signal for print-outputting. At this time, the optimum binarization method varies depending upon the attribute of the image. More specifically, for graphics such as characters and patterns, binarization, in which the size of the dither matrix is made smaller to place priority on resolution, is preferable. Also, for such a natural image as a photograph, it is preferable to make the matrix size larger for placing priority on tonal reproducing ability.

In order to switch the binarization method in accordance with the image as described above, there can be also conceived such a method that the rasterizer 143 does not convert image data into CMYK, but develops the image data into the bitmap image using the values of RGB, the image forming unit detects a character image area from among the received bitmap images of RGB using the known image area separation method, and switches the converting method of converting RGB to CMYK in the detected character image area and in any areas other than it while generating CMYK data for outputting.

In this case, however, there is a drawback in that the image area separation method is not reliable because it is not always capable of detecting the character area with 100% reliability, but may erroneously detect a part of the natural image area as a character area.

In contrast, according to the present embodiment, it is possible to simply and reliably apply the optimum binarization method to each image because the binarization method is switched by using the attribute map information. For example, in Fig. 4, the 0th bit representing the attribute map information corresponding to each pixel of the character image (b) and the graphic image (a) (Vector) is "0", and the 0th bit representing the attribute map information

corresponding to each pixel of the natural image (c)

(bitmap) is "1". For this reason, the natural image

can be simply cut and separated from the character

image and graphic image for each pixel.

5   And, for a pixel having 1 as the 0th bit in the

attribute map, the dither matrix shown in Fig. 21 is

used, and for a pixel having 0 as the 0th bit in the

attribute map, the dither matrix shown in Fig. 22 is

used. The multi-value data used in the present

10   embodiment has 8 bits for representing 0 to 255 levels.

When the value of multi-value data of each pixel is

larger than the threshold value of each cell of the

dither matrixes of Figs. 21 and 22, the bit is turned

ON, and when it is equal to or smaller than the

15   threshold value of each cell, the bit is turned OFF.

Fig. 21 is a view showing an example of dither

matrix having a size of 3 × 3. According to the dither

matrix shown in Fig. 21, assuming an original image of

600 dpi, it can be reproduced in resolution of 200 dpi,

20   and the number of gradation levels is 10. Although the

number of gradation levels is small, it is possible to

reproduce the image with high resolution. Fig. 22 is a

view showing an example of dither matrix having a size

of 8 × 8. According to the dither matrix shown in Fig.

25   22, assuming an original image of 600 dpi, it can be

reproduced at resolution of 75 dpi, and the number of

gradation levels is 65. Although the resolution
becomes low, the number of gradation levels is great,
and the natural image can be reproduced at higher
quality than when the dither matrix of Fig. 21 is used.

5   As described above, according to the second
embodiment, since the dither matrix is switched for
each pixel on the basis of the attribute map
information, it becomes possible to reproduce the
character and graphic at high resolution while the
10   gradation is retained in the natural image, and to
provide a high-quality image.

[Third Embodiment]

        In the second embodiment, the description has been
15   made of a case where there is 1-bit information as the
attribute map. However, the attribute information for
each pixel of the attribute map is not limited to 1-bit.
In a third embodiment, as an example of attribute
information including a plurality of bits, the
20   description will be made of a case where 2-bit
attribute information has been adopted.

        Fig. 25 is a view showing an example of a data
format for attribute information according to the third
embodiment. This has 2-bit information consisting of
25   0th bit and 1st bit, and the 0th bit is a bitmap flag,
and is the same as described in the second embodiment.

More specifically, if the bitmap flag is "1", it is shown to be a pixel generated from the bitmap object, and if "0", it is a pixel generated from the vector object, that is, it is shown to be a pixel in the character or graphic image.

The 1st bit of the attribute map is a color flag, and if it is 1, it shows color, and if it is 0, it shows monochrome.

In this respect, in the present embodiment, the 0th bit and the 1st bit are independent, and switching of the dither matrix as shown in the second embodiment is performed using the information of the 0th bit. Thus, by referring to the information of the 1st bit, it becomes possible to further provide optimum image processing.

For example, in a case where color shift is remarkable because of the characteristics of the image forming unit 149, when a character object having no color is outputted as a combination of four colors of C, M, Y and K, a color appears at the edge portion of a black character because of the color shift, and it becomes unsightly. In order to solve this problem,

$$C' = C-min\ (C,\ M,\ Y)$$

$$M' = M-min\ (C,\ M,\ Y)$$

$$Y' = Y-min\ (C,\ M,\ Y)$$

$$K = min\ (C,\ M,\ Y)$$

C′, M′, Y′ and K are newly prepared as shown above, and a process to replace any color with K (black) as far as possible, so-called UCR (Under Color Removal) process is used. In this case, color (C, M, Y) information

5   does not spread well on a monochrome character image, and the color shift does not become conspicuous. If black spreads much on a natural image conversely, there is the problem in that a dark color with lowered chroma appears and an unsightly image is generated depending

10   upon the characteristics of the image forming unit 19. It is difficult to cause both a character image and a natural image to have high image quality.

In contrast, according to the third embodiment, it becomes possible to have both character images and

15   natural images to be subjected to appropriate image processing. Through the use of the 2-bit attribute map shown in Fig. 25, each pixel is to have any of the following attribute information:

0 bit	1 bit	
0	0	monochrome in vector
0	1	color in vector
1	0	monochrome in bitmap
1	1	color in bitmap

In case of "monochrome in vector", color shift is prevented by applying the UCR process described above, and in case of "color in bitmap", it is possible to realize a high-quality image with high chroma by stopping the UCR process so as not to spread K (black) as far as possible. In this respect, such color processing is properly performed in the step S23 in the flow chart shown in Fig. 24, by switching a conversion process from RGB data to YMCK data in accordance with the attribute information read in the step S22.

In this case, it is important that the attribute map be capable of coping with a low-priced printer whose function is limited to a high value-added printer having more functions by having information hierarchically. For example, by increasing the number of bits of the attribute map from 1 to 2, it becomes possible to carry out either the second embodiment or the third embodiment. Thus, if two bits are used as in the third embodiment, that is, if the number of bits is increased, it becomes possible to switch image processing, making it possible to provide a higher quality image.

Therefore, it is possible to perform the processing, with a low-priced printer whose function is limited, by using only the lower bit of the attribute

information, and to perform highly developed processing, with a printer having more functions, by using more bits of the attribute information. In this manner, it is possible to arrange the configuration so that

5 attribute information which is more frequently used basically located in the lower bits and attribution information is hierarchically retained, whereby a more appropriate configuration in accordance with the characteristics (such as attaching importance to cost

10 or image quality) of the system is constructed by using the same structure of the attribute map.

[Fourth Embodiment]

In the third embodiment, description is made of a

15 case where each of a plurality of bits of attribute information is independent of one another, but the plurality of bits of attribute information may be related to one another. In a fourth embodiment, description will be made of a case where a plurality of

20 bits of attribute information are in a master and servant relationship to one another.

Fig. 26 shows an example of a data format for attribute information according to the fourth embodiment. This attribute information has 3-bit

25 information consisting of 0th bit, 1st bit and 2nd bit. The 0th bit is a bitmap flag, and as in the case of the

second and third embodiments, "1" indicates a pixel generated from the bitmap object, while "0" indicates a pixel generated from the vector object, that is, a character or graphic image.

Also, the 1st bit is a color flag, and as in the case of the third embodiment, "1" indicates color while "0" indicates monochrome (white having no color information, gray, black).

The 2nd bit is a character flag, and "0" indicates any object other than characters while "1" indicates a character object.

When the information up to the 2nd bit of the attribute map is used, the following becomes possible. For example, in a case where the color of TEXT (b) is defined as black of R=G=B=0 in Fig. 4, when the optimum CMYK signal for this is represented by a 8-bit density signal, the following equations are established: C=M=Y=0 and K=255. In other words, a black character is preferably reproduced only by black toner of four colors of toners for the printer.

On the other hand, in a case where the pixel value of a specific pixel in the natural image (c) is R=G=B=0, if this is converted into C=M=Y=0 and K=255 as in the case of the character data, it becomes low in absolute density because reproduction of a portion originally having the highest density of the natural image is

performed only by black toner. In this case, therefore, if it is converted into a value such as C=M=Y=100 and K=255 to increase the absolute density, desirable results will be obtained.

5   As to the graphic image (a), it is assumed that an intermediate value between the character and the natural image is optimum. That is, it is assumed that C=M=Y=50 and K=255 are the optimum values for R=G=B=0. In this case, there can be also conceived a method to
10 subject a multi-value bitmap image to be transferred from the image memory to an image area separation process for appropriately switching color processing (color processing for R=G=B=0) as described above. However, the same drawback (the image area separation
15 method is not reliable because it is not always capable of 100% detecting the character area, but may erroneously detect a part of the natural image area as a character area) as described above cannot be prevented.

20   In contrast, if attribute map information according to the fourth embodiment is used, the character image (b) is configured by pixels whose 0th bit and 2nd bit are set to 0 and 1 respectively in the attribute map, the graphic image (a) is configured by
25 pixels whose 0th bit and 2nd bit are set to 0 in the attribute map, and the natural image (c)(bitmap) is

configured by a pixel whose 0th bit is set to 1 in the attribute map. For this reason, the natural image, the character image and the graphic can be simply separated for each pixel. Therefore, as described above, it

5. becomes possible to simply and accurately realize that pixel value of R=G=B=0 is converted into C=M=Y=0 and K=255 in the character image unit, pixel value of R=G=B=0 is converted into C=M=Y=50 and K=255 in the graphic image unit, and pixel value of R=G=B=0 is

10 converted into C=M=Y=100 and K=255 in the natural image unit, thus making it possible to realize better image quality. In this respect, the above-described processing for black can be realized by switching the color processing in the step S23 on the basis of the

15 attribute information read in the step S22 of Fig. 24.

As described above, it is possible to provide finer attribute by a combination of a plurality of bits.

The difference between the fourth embodiment and the third embodiment is as follows:

20 The information of the 0th bit and the information of the 1st bit are independent respectively, and can be independently used, whereas the information of the 2nd bit is to be used in a combination with the information of the 0th bit. This time, since two bits of the 0th

25 bit and the 2nd bit are used, it is possible to provide four types (the square of 2) of attributes. It is

- 39 -

possible to provide eight types of attributes for combinations for three bits, and to provide 16 types of attributes for combinations for four bits.

5   When the 0th bit and the 2nd bit are both 1 as shown below, it indicates the ground (state in which there is no object information). When there are three (natural image, character image and graphic) object information, there can be provided four types of information for two bits, and therefore, it is possible

10   to provide quite different attributes. Although the attribute of a character in a bitmap is inconsistent, the information of the 2nd bit cannot be independently used, but is handled such that it is used as subordinate information having different meaning

15   depending upon the information of the 0th bit.

0 bit	2 bit	
0	0	other than character & vector = Graphic
0	1	character & vector = Character
1	0	other than character & bitmap = Bitmap
1	1	character & bitmap = Ground

In the fourth embodiment, the description is made of an attribute map by a combination of a plurality of

bits. Although the 0th bit is independent information, the 2nd bit has an attribute subordinate to the 0th bit, is used as a combination of two bits. Thus, an attribute is obtained by combining a plurality of bits.

5   In this respect, it is possible to make the attribute due to a combination of a plurality of bits have no independent attribute. That is, it is possible to configure it such that it doesn't make any sense if not combined. As shown in, for example, Fig. 27, it is

10 possible, in the configuration of the 0th bit and the 1st bit, to configure the attribute information as shown below.

0-bit	1-bit	
0	0	Ground
0	1	Graphic
1	0	Character
1	1	Bitmap

15   In any case, it becomes possible to improve processing speed by omitting the process for a pixel whose attribute is the ground.

[Fifth Embodiment]

- 41 -

Fig. 28 is a sectional view showing the internal structure of a laser beam printer (hereinafter, abbreviated to LBP), which is applied to the present embodiment, and this LBP is capable of registering a character pattern, a repetitive format (form data) or the like from a data source (not shown) such as a host computer.

In Fig. 28, the reference numeral 1000 denotes the LBP body, which inputs and stores character information (character code), form information or macro instruction or the like, which are supplied from a host computer connected to the outside, and prepares a character pattern, a form pattern or the like which correspond in response to those information to form an image on a recording sheet, which is a recording medium. The reference numeral 1012 denotes a control panel equipped with a switch for operation, a LED display or the like; 1001, a printer control unit for controlling the entire LBP1000 and analyzing character information or the like to be supplied from the host computer. This control unit 1001 mainly converts the character information into a video signal of the character pattern corresponding thereto to output to a laser driver 1002. The laser driver 1002 is a circuit for driving a semiconductor laser 1003, and switches ON-OFF laser light 1004 to be emitted from the semiconductor laser

1003 in accordance with a video signal inputted. The laser 1004 is scanned in the left and right directions by a polygonal rotating mirror 1005 to scan on an electrostatic drum 1006. This forms an electrostatic

5   latent image for a character pattern on the electrostatic drum 1006. This latent image is formed for each of yellow, magenta, cyan and black colors, and these latent images are developed by yellow, magenta, cyan and blank developer units 1007 around the

10   electrostatic drum 1006 respectively, and thereafter, are transferred onto a recording sheet. For this recording sheet, a cut sheet is used, and cut sheet recording sheet is contained on a sheet cassette 1008 mounted to the LBP1000, and is taken into the apparatus

15   by means of a feed roller 1009 and conveying rollers 1010 and 1011 to be supplied to the electrostatic drum 1006.

In a printer system thus configured, with reference to Figs. 29 and 30, the description of the

20   fifth embodiment will be made in accordance with the flow chart of Fig. 31.

In Fig. 29, the reference numeral 301 denotes a PDL interpretation unit for interpreting PDL (Page Description Language: for example, post-script,

25   registered trade-mark, GDI: registered trade-mark or the like) data; 302, an object to be prepared by the

PDL interpretation unit 301; 303, a graphic library for converting the object 302 into an instruction 304 which a rendering engine can interpret and adding object information (Flag) 305 (For the object information, see

5   Fig. 31 to be described later); 306, a rendering engine; 307, a bitmap image of (8 bits for each of RGB colors + object information bit (Flag) 3 bits)/pixel; and 308, a pre-processing module for performing image processing for printing the bitmap image 307 with a

10 printer engine 309.

Fig. 30 shows an object information 305 in detail. The object information uses 3 bits, 0th bit is a bit (will be called "Bitmap flag") showing that the object is a bitmap object or a vector graphic, and 1st bit is

15 a bit (will be called "color flag") showing that the object is a color object or a monochrome object. 2nd bit is a bit showing that the object is a character or not when the bitmap flag is 0 (=Bitmap data), and a bit (will be called "character flag") showing gradation

20 priority or resolution priority when the bitmap flag is 1 (=vector data).

Hereinafter, with reference to Fig. 32, the description will be made with respect to the procedure in processing blocks shown in Fig. 29.

25 When the LBP1000 receives print data(step S502), the PDL interpretation unit 301 interprets the data

(step S503). The PDL interpretation unit 301 outputs the object 302 with the interpretation result as one rendering instruction of one object in most cases (step S504). The graphic library 303 interprets the object 302 (step S505), and prepares the instruction 304 by adding the object information flag 305 on the basis of the interpretation result (step S506). The rendering 306 renders the instruction 304 and the object information flag 305 to prepare the bitmap image 307 (step S507). At this time, the rendering object is only the instruction 304, and the object information flag 305 is passed through as it is to be added to each pixel of the rendering result. The pre-processing unit 308 subjects the bitmap image 307 to pre-processing for printing by the engine 309 (step S508). Concretely, RGB24Bit → CMYK1Bit conversion (1-color conversion, binarization) and black character treatment (in case of a color character whose pixel value gets closer and closer to black, more and more a process of painting the character uses black toner) or the like are performed. The engine 309 prints the result (step S509) to complete the processing.

In the above-described configuration, the rendering result of the rendering engine has been set to 8 bit each of RGB /Pixel, but it goes without saying

that the similar result can be obtained for the
rendering engine which renders in the other color space.

As described above, according to the present
embodiment, even after the completion of the rendering,

5   it is possible for the pixel to retain the information
of the original object, and more accurate post-
processing such as edge extraction can be performed.

In this respect, the present embodiments may be
applied to a system configured by a plurality of

10   equipment (such as, for example, a host computer, an
interface equipment, a reader and a printer) and to an
apparatus configured by one equipment.

Also, it goes without saying that the objects of
the above-described embodiments can be achieved even if

15   storage media, in which program codes for software for
realizing the functions of the embodiments have been
recorded, are supplied to the system or the apparatus,
and the computer (or CPU or MPU) of the system or the
apparatus reads out and executes the program codes

20   stored in the recording media.

In this case, the program codes themselves read
out from the storage media are to realize the functions
of the embodiments described above, and the storage
media in which the program codes have been stored are

25   to configure the present invention.

For the storage medium for supplying the program code, there can be used, for example, a floppy disk, a hard disk, an optical disk, a magneto-optic disk, CD-ROM, CD-R, magnetic tape, a non-volatile memory card, ROM or the like.

Also, it goes without saying that there is included a case where a program code read out by the computer is executed, whereby the function of the above-described embodiment is not only realized, but also OS (operating system), which is operating on the computer, or the like perform a part or the whole of the actual processing on the basis of the instruction of the program code to realize the function of the embodiment described above by the processing.

Further, it goes without saying that there is also included a case where the program code read out from the storage medium is written in a function expanded board inserted into the computer or in a memory provided for a function expansion unit connected to the computer, and thereafter, the CPU or the like provided for the function expanded board or the function expansion unit perform a part or the whole of the actual processing on the basis of the instruction of the program code to realize the function of the embodiment described above by the processing.

As described above, on print-outputting image data, appropriate data processing based on the attributes of each image included in the image data is realized and the quality of the output image is improved.

5   As many apparently widely different embodiments of the present invention can be made without departing from the spirit and scope thereof, it is to be understood that the invention is not limited to the specific embodiments thereof except as defined in the
10   appended claims.